

INICIAR SESIÓN

NUESTROS PLANES

TODOS LOS  
CURSOS

FORMACIONES

CURSOS

PARA  
EMPRESAS

ARTÍCULOS DE TECNOLOGÍA &gt; DATA SCIENCE

# SQL JOIN: Aprenda INNER, LEFT, RIGHT, FULL e CROSS



danielleoliveira046

14/02/2022



Esse artigo faz parte da  
Formação Data Science



```
SELECT <select_list>  
FROM Tabela A  
INNER JOIN Tabela B  
ON A.Key = B.Key
```

[



En el lenguaje **SQL**, ejecutamos diferentes consultas para generalmente crear informes que serán posteriormente utilizados para análisis y la tomada de alguna decisión, y para crear consultas cada vez más completas necesitamos utilizar varias tablas en conjunto y para eso usamos los **JOINS**.

## Que es y para qué sirve Join en SQL?

Una cláusula **JOIN** en SQL, correspondiente a una operación de conjunción en álgebra relacional, combina columnas de una o más tablas en un **banco de datos relacional**. Ella

crea un conjunto que puede ser salvo como una tabla o usado de la forma como está.

Un JOIN es un medio de combinar columnas de una o más tablas, usando valores comunes a cada una de las. El estándar SQL [ANSI](#) especifica cinco tipos de JOIN: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN e CROSS JOIN.

Como un caso especial, una tabla (tabla base, visión o tabla juntada) puede juntarse a sí misma en una auto-uniión (self-join).

En un banco de datos relacional, los datos son distribuidos en varias tablas lógicas. Para obtener un conjunto completo y significativo de datos, es necesario consultar datos de esas tablas usando junctiones (JOINS).

Como ya fue mencionado arriba, existen diferentes tipos de JOINS en el SQL. Vamos a definir y ejemplificar cada uno de ellos.

## INNER JOIN

```
SELECT <select_list>
FROM Tabla A
INNER JOIN Tabla B
ON A.Key = B.Key
```

La cláusula **INNER JOIN** compara cada línea de tabla A con las líneas de tabla B para encontrar todos os pares de líneas que satisfacen la condición de junción. Si la condición de junción fuera evaluada como TRUE, los valores de la columna de las líneas correspondientes de las tablas A y B serán combinadas en una nueva línea e incluidos en el conjunto de resultados.

## LEFT JOIN

Retorna todos los registros de la tabla izquierda y los registros correspondientes de la tabla derecha.

```
SELECT <select_list>
FROM Tabla A
LEFT JOIN Tabla B
ON A.Key = B.Key
```

Para cada línea de la tabla A, la consulta la compara con todas las líneas de la tabla B. Si un par de líneas hiciera con que la condición de unión sea evaluada como TRUE, los valores de la columna de esas líneas serán combinadas para formar una nueva línea que será incluida en el conjunto de resultados.

Si una línea de la tabla **“izquierda” A** no tuviera ninguna línea correspondiente de la tabla **“derecha” B**, la consulta irá combinar los valores de la columna de la línea de la tabla **“izquierda” A** con NULL para cada valor de la columna de la tabla de la **“derecha” B** que no satisface la condición de junto (FALSE).

En resumen, la cláusula LEFT JOIN retorna todas las líneas de la tabla **“izquierda” A** y las líneas correspondientes o valores NULL de la tabla **“izquierda” A**.

## RIGHT JOIN

Retorna todos los registros de la tabla derecha y los registros correspondientes de la tabla izquierda.

```
SELECT <select_list>
FROM Tabla A1
ON A.Key = B.Key
```

La RIGHT JOIN combina datos de dos o más tablas. La RIGHT JOIN comienza a seleccionar datos de la tabla **“derecha” B** y a corresponder las líneas de la tabla **“izquierda” A**.

La RIGHT JOIN retorna un conjunto de resultados que incluye todas las líneas de la tabla **“derecha” B**, con o sin líneas correspondientes en la tabla **“izquierda” A**. Se una línea en la tabla **derecha B** no tuviera ninguna línea correspondiente de la tabla **“izquierda” A**, la columna de la tabla **“izquierda” A** en el conjunto de resultados será nula igualmente a lo que acontece en el LEFT JOIN.

## FULL JOIN

Retorna todos los registros cuando hubiera una correspondencia en la tabla izquierda o derecha.

```
SELECT <select_list>
FROM Tabla A
```

```
FULL JOIN Tabla B  
ON A.Key = B.Key
```

La cláusula `FULL JOIN` retorna todas las líneas de las tablas unidas, correspondidas o no, o sea, tú puedes decir que la `FULL JOIN` combina las funciones de la `LEFT JOIN` y de la `RIGHT JOIN`. `FULL JOIN` es un tipo de junción externa, por eso también es llamada junción externa completa.

Cuando no existen líneas correspondientes para la línea de la tabla izquierda, las columnas de la tabla derecha serán nulas. De la misma forma, cuando no existen líneas correspondientes para la línea de la tabla derecha, la columna da tabla izquierda será nula.

## CROSS JOIN

```
SELECT <select_list>  
FROM Tabla A  
CROSS JOIN Tabla B
```

La cláusula `CROSS JOIN` retorna todas las líneas de las tablas por cruzamiento, o sea, para cada línea de la tabla izquierda queremos todas las líneas de la tabla derecha o viceversa. Él también es llamado de producto cartesiano entre dos tablas. Pero, para eso es necesario que ambas tengan el campo en común, para que la conexión exista entre las dos tablas.

Para entender mejor, piense que tenemos un banco de datos, donde tenemos una tabla `FUNCIONARIO` y una tabla `CARGO`, las podríamos tener varios cargos para un único `FUNCIONARIO`, y usando el `CROSS JOIN` podemos traer todos los `CARGOS` de todos los `FUNCIONARIOS`.

## Conclusión

Vimos que los **JOINS** en el lenguaje **SQL** son muy importantes, ellos nos permiten crear consultas más completas y más complejas de forma simple, utilizando pruebas lógicas (condición de junción), que son de extrema importancia cuando trabajamos con bancos de datos.

¿Te gusto el artículo y quieres saber más sobre SQL y JOINS? La [Formación de SQL con MySQL](#) te va a enseñar como crear JOINS y mucho más.



Esse artigo faz parte da  
Formação Data Science

Cursos de Data Science

ARTÍCULOS DE TECNOLOGÍA > DATA SCIENCE

**En Alura encontrarás variados cursos sobre Data Science.  
¡Comienza ahora!**

**SEMESTRAL**

**US\$49,90**

un solo pago de US\$49,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 6 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**

[Paga en moneda local en los siguientes países](#)

**ANUAL**

**US\$79,90**

un solo pago de US\$79,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 12 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**

[Paga en moneda local en los siguientes países](#)

Acceso a todos  
los cursos

Estudia las 24 horas,  
dónde y cuándo quieras

Nuevos cursos  
cada semana

**NAVEGACIÓN**

PLANES



INSTRUCTORES

BLOG

POLÍTICA DE PRIVACIDAD

TÉRMINOS DE USO

SOBRE NOSOTROS

PREGUNTAS FRECUENTES

## ¡CONTÁCTANOS!

¡QUIERO ENTRAR EN CONTACTO!

## BLOG

PROGRAMACIÓN

FRONT END

DATA SCIENCE

INNOVACIÓN Y GESTIÓN

DEVOPS

AOVS Sistemas de Informática S.A  
CNPJ 05.555.382/0001-33

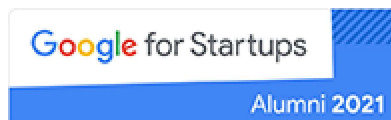
## SÍGUENOS EN NUESTRAS REDES SOCIALES



## ALIADOS



En Alura somos unas de las Scale-Ups seleccionadas por Endeavor, programa de aceleración de las empresas que más crecen en el país.



Fuimos unas de las 7 startups seleccionadas por Google For Startups en participar del programa Growth Academy en 2021

## CURSOS

### Cursos de Programación

Lógica de Programación | Java

### Cursos de Front End

HTML y CSS | JavaScript | React

### Cursos de Data Science

Data Science | Machine Learning | Excel | Base de Datos | Data Visualization | Estadística

### Cursos de DevOps

Docker | Linux

### Cursos de Innovación y Gestión

Productividad y Calidad de Vida | Transformación Ágil | Marketing Analytics |  
Liderazgo y Gestión de Equipos | Startups y Emprendimiento